git config --global user.name “Your Name”

global表示全局，所有的仓库都会使用这个配置，可以不使用它给某个仓库单独指定不同的配置。

git config --global user.email “xxx@example.com”

git config –l 查看配置内容

创建版本库

git init 创建版本库

版本提交

git add readme.txt

git commit –m “worte a readme file”

git status

git diff readme.txt

版本回退

git log

git log –pretty=oneline

git reset --hard HEAD^ 回退到上个版本

git reset --hard HEAD^^ 回退到上上个版本

git reset --hard 1094a 回退到特定版本

git reflog 记录你的每一次命令

工作区：电脑里的目录

版本库：.git目录是Git的版本库，里面存了很多东西，其中最重要的就是称为stage的暂存区，还有Git为我们自动创建的第一个分子master，以及指向master的第一个指针叫HEAD。

git add就是把文件的修改添加到暂存区。

git commit把暂存区的所有内容提交到当前分支。

撤销修改

git checkout –readme.txt 丢弃工作区的修改

git reset HEAD readme.txt 把暂存区的修改撤销掉，重新放回工作区

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD <file>，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。

删除文件

rm test.txt

git rm test.txt

git commit –m “romve”

远程仓库

ssh-keygen –t res –C [youremail@example.com](mailto:youremail@example.com) 用户主目录下.ssh目录内创建SSH Key

在Github上Add SSH Key

git remote add origin [git@git.com:xuechencheng/learngit.git](mailto:git@git.com:xuechencheng/learngit.git) 关联Github

git push –u origin master origin是远程仓库名，第一次需要使用-u

git remote –v 查看远程库信息

git remote rm origin 解除本地和远程库的绑定关系

git clone git@github.com:xuechencheng/gitskills.git

分支管理

HEAD不是指向提交，而是指向master，master才是指向提交的，HEAD指向的就是当前分支。创建新的分支dev后，HEAD就会指向dev。合并dev分支就是把master指向dev的当前提交。

git checkout –b dev 创建并切换到dev分支，相当于以下两条指令

git branch dev

git checkout dev

git branch 查看当前分支

git merge dev把dev分支的工作合并到master分支上

git branch –d dev删除分支

git switch –c feature1 使用switch代替checkout

解决冲突

git merge feature1

手动修改冲突内容，然后使用git add。

git log --graph --pretty=oneline --abbrev-commit 查看分支合并情况

git merge --no-ff -m "merge with no-ff" dev 禁用fast-forward，留下合并日志

Bug分支

每个bug都可以通过一个新的临时分支来修复

git stash 把当前工作现场储藏起来

git stash list查看工作现场

git stash pop 恢复的同时把stash内容也删除

git stash apply stash@{0} 恢复但不删除

git cherry-pick 4c805e2 复制一个特定的提交到当前分支

git branch –D feature 强行删除一个还没有合并的分支

多人协作

当你从远程仓库克隆时，实际上Git自动把本地的master分支和远程的master分支对应起来了，并且，远程仓库的默认名称是origin。

git remote 查看远程库的信息

git remote –v 更详细的信息

git push origin dev 推送分支

git checkout –b dev origin/dev 创建远程的origin的dev分支到本地

git branch –set-upstream-to=origin/dev dev指定本地分支与远程origin/dev分支的链接

git rebase 把分叉的提交历史整理成一条直线

标签管理

标签是版本的一个快照。标签不能移动，创建和删除标签都是瞬间完成的。

git tag v1.0 打标签

git tag 查看标签

git tag v0.9 f52c633 给特定版本打标签

git show v0.9

git tag –a v0.1 –m “version 0.1 released” 1094adb

git tag –d v0.1 删除标签

git push origin v1.0 推送标签

git push origin –tags 推送所有标签

git tag –d v0.9; git push origin :refs/tags/v0.9 删除远程标签

自定义Git

git config --global color.ui true

git config –global alias.st status 别名

忽略特殊文件

在Git工作区的根目录下创建一个特殊的.gitignore文件，然后把要忽略的文件名填进去。

全局配置文件在哪里？.git/config中。